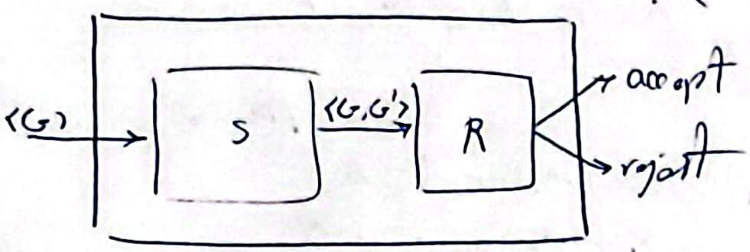
محمد رضا امانلو ۸۱۰۱۰۰۰۸۲ تکلیف یازدهم درس نظریه اسفندیار

۱) می‌دانیم زبان All_{CFG} یک زبان تصمیم ناپذیر است! حال فرض می‌کنیم EQ_{CFG} تصمیم پذیر باشند (فرض خلف)
فرض می‌کنیم ماشین R، تصمیم گیرنده (decider) زبان EQ_{CFG} است. مسئله All_{CFG} را به مسئله EQ کاهش
می‌دهیم. درواقع ($All_{CFG} \leq EQ_{CFG}$)!

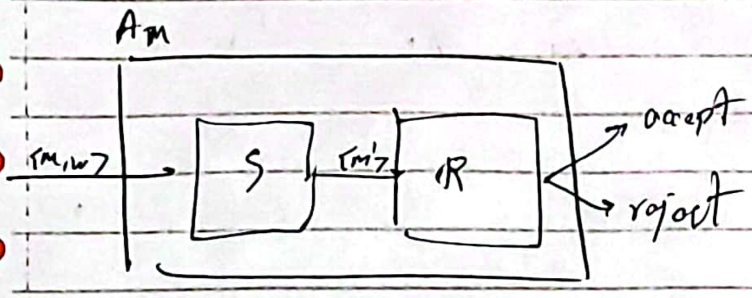


در S گرامر G' را مطابق زیر ایجاد می‌کنیم.

$G': S \rightarrow 0 \mid 1 \mid \varepsilon \mid 0S \mid 1S$

حال R را روی <G, G'> اجرا می‌کنیم، درصورتی که accept برگرداند، All_{CFG} هم accept و در غیر این صورت reject می‌کند
با فرض تصمیم پذیر بودن EQ_{CFG} توانستیم برای All_{CFG} یک decider بسازیم که تناقض است پس
فرض خلف نادرست بوده پس EQ_{CFG} تصمیم پذیر نیست! ✓

۲) $(A_{TM} \leq R)$



باید A_{TM} را به R کاهش دهیم.

فرض میکنیم که R که مسئله‌ی صورت سوال را decide می‌کند وجود دارد (فرض خلف)

در S باید M' را بسازیم. (از روی <M,w>)

تعریف M':

M' برای هر ورودی M را روی نوار اول خود شبیه‌سازی می‌کند، درصورتی که accept شد روی نوار دوم خود یک حرف non-blank را می‌نویسد.
در غیر این صورت هیچ چیز روی نوار دوم خود نمی‌نویسد!

سپس M' را به R (به عنوان decider برای زبانی که اگر روی نوار دوم ماشین تورینگ نوشته بشود، accept می‌کند) می‌دهیم، ما فرض کردیم R decider است! پس <M'> را میگیرد و یکی از ۲ حالت accept یا reject را خروجی می‌دهد. پس A_{TM} را decide می‌کند که این تناقض است. ✗

پس فرض خلف نقض می‌شود پس R یا همان مسئله‌ی صورت سوال تصمیم ناپذیر است ✓

۳) حالت‌بندی می‌کنیم (همه‌ی حالت‌ها)

حالت ۱) بلاک A در همه‌ی دومینوها، اندازه‌ای متفاوت با B باشد. در این صورت مسئله جوابی ندارد و باید reject کنیم.

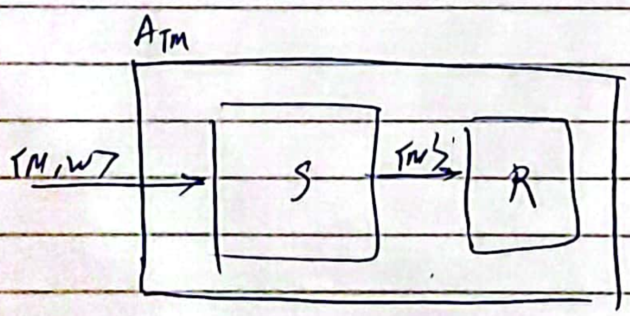
حالت ۲) هر دو بلاک دومینو هم‌سایز باشند. در این صورت همین یک دومینو جواب است پس accept می‌کنیم.

حالت ۳) در بعضی دومینوها بلاک A و در بعضی B بلندتر باشند. مثلاً اگر در دومینوی طول بلاک A، n تا از طول بلاک B آن دومینو طولانی‌تر باشد و در دومینوی دیگر طول بلاک A، ۲ تا از طول بلاک B کوتاه‌تر باشد، حال با چیدن ۲ تا از دومینوی اول در کنار n تا از دومینوی دوم، یک رشته‌ی درست و حالت قابل قبول می‌سازیم.

پس همه‌ی موارد و تمامی حالت‌های ممکن یا به accept رسیدیم و یا به reject پس مسئله decidable است.

۴) فرض می‌کنیم T یک زبان تصمیم‌پذیر است و R را به عنوان decider آن در نظر می‌گیریم.

حال A_{TM} را به R کاهش می‌دهیم: $A_{TM} \le R$



در S باید M' را به نحو زیر بسازیم:

برای ورودی x در M' اگر x، 01 یا 10 را نداشت reject می‌کنیم. در غیر این صورت اگر فرم 01 را داشت accept می‌کنیم و اگر فرم 10 را داشت، M را روی w اجرا می‌کنیم و پاسخ را برمی‌گردانیم.

x has the form 01 or 10
- no → reject
- yes →
  - form 10, if M accepts w accept else reject
  - form 01, accept

(۵) فرض می‌کنیم $AMBIG_{CFG}$ تصمیم‌پذیر است (فرض خلف). سپس PCP را به آن کاهش می‌دهیم و به تناقض می‌رسیم. ($PCP \le AMBIG_{CFG}$)

CFG زبانی مبهم است که PCP یک match داشته باشد. در CFG دیدیم که $S \to A|B$ را به عنوان شروع ایجاد می‌کنیم. حال تمامی مسیرهای ممکن از A و B را می‌نویسیم. در واقع اگر PCP جواب داشته باشد بدین معنی است که $t_1t_2t_3 \dots = b_1b_2 \dots b_k$ شده است. در واقع،

$A \to$ ◯1 | ◯2 | ◯3 | — | ◯n
◯A1 | ◯A2 | ◯A3 | — | ◯An

که شماره‌های نوشته شده به معنی شماره دومینو استفاده شده است.

در واقع چون CFG به صورت مقابل است

$S \to S_1 \mid S_2$

$S_1 \to t_1 S_1 a_1 \mid t_2 S_2 a_2 \mid \dots \mid t_1 a_1 \mid t_2 a_2 \mid \dots \mid t_k a_k$

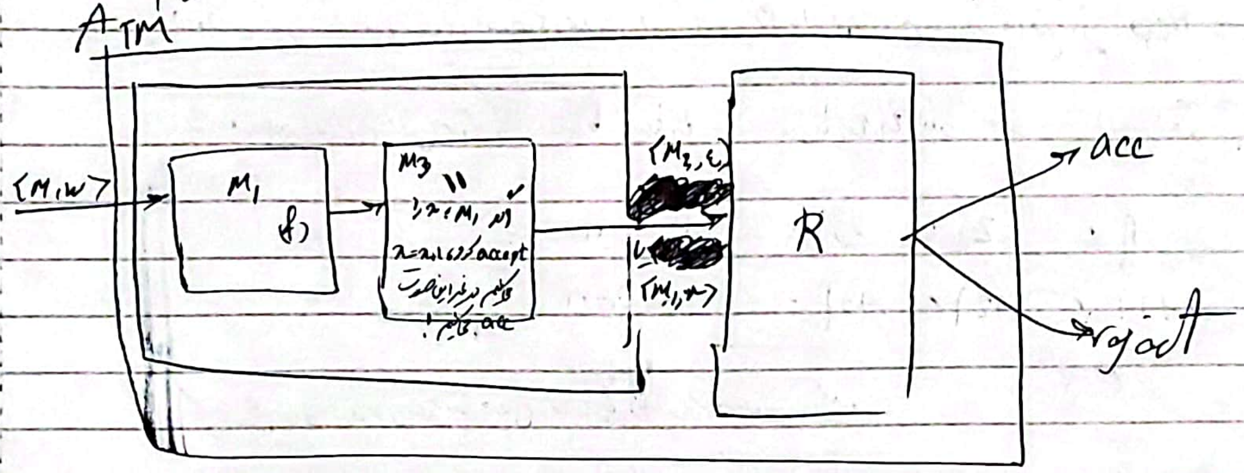
$S_2 \to b_1 S_1 a_1 \mid b_2 S_2 a_2 \mid \dots \mid b_2 a_1 \mid b_2 a_2 \mid \dots \mid b_k a_k$

حال اگر دومینوهایی به شکل $\left(\frac{t_1}{b_1}\right) - \left(\frac{t_k}{b_k}\right)$ درست کنیم. به ازای هر matching که در PCP بتوانیم ایجاد کنیم با شروع از یکی از دو variable مختلف S_1 و S_2 توانسته‌ایم به همان رشته معین برسیم پس توانسته‌ایم با دو درخت اشتقاق متفاوت رشته را بسازیم پس گرامر مبهم است! با این کار توانستیم نشان دهیم که مسئله حل‌نشدنی PCP را می‌توانیم روی مسئله حل‌نشدنی $AMBIG_{CFG}$ بیاندازیم

پس $AMBIG_{CFG}$ مسئله‌ای تصمیم‌ناپذیر است. (هرگاه گرامر مبهم است یک matching رخ داده است و هرگاه مبهم نباشد matching رخ نمی‌دهد)

۶) اگر ماشین تورینگ M_1 را ماشینی درنظر بگیریم که تابع f_1 را روی ورودی خود انجام می‌دهد. و ماشین تورینگ M_2 را ماشینی درنظر بگیریم که به M_1 همه‌ی اعداد طبیعی را ورودی می‌دهد.

با برهان خلف ثابت می‌کنیم. ($A_{TM} \leq R$) R را یک decider برای مسئله فرض می‌کنیم.



M_1، f_1 را روی w اجرا می‌کند، سپس M_3 برای همه‌ی رشته‌های خروجی M_1، عدد x را به M_1 می‌دهد اگر M_1 آن را accept کرد آن را به اضافه یک می‌کند و مجدداً به خود M_1 می‌دهد
تا وقتی که M_1، accept نکند. در این حالت در واقع ما به یک شکل نقض رسیده‌ایم!

* در واقع M_1، تنها f را اجرا می‌کند و M_2، M_1 را روی اعداد طبیعی اجرا می‌کند. (برای ورودی x اگر پذیرش M_1 را روی x+1 صدا می‌زند) حال اگر شکل نقض وجود داشته باشد M_2 halt نمی‌کند برای حل این مشکل R را داریم!

* M_2 برای ورودی w برای هر عدد طبیعی x، R را روی ⟨M_1,x⟩ اجرا می‌کند، اگر R ریجکت کرد، M_2 accept می‌کند و در غیر این صورت halt نمی‌کند. اما ما باید بدانیم که شکل نقض داریم یا نه؟ اگر نداشته باشد M_2 halt نمی‌کند که مجدداً با استفاده از همین روش و R مشکل را حل می‌کنیم.

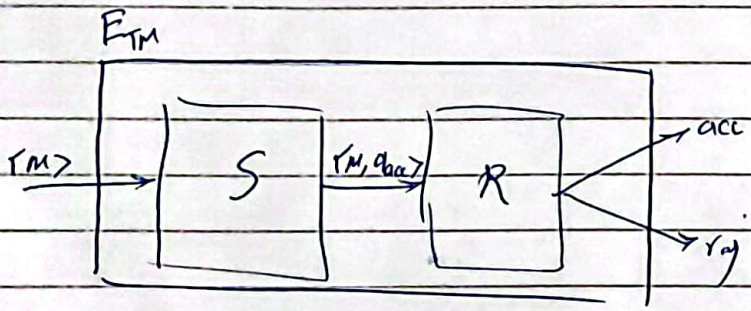
* M_3 برای ورودی w، R را برای ⟨M_2,ε⟩ اجرا می‌کند که حتماً پایان یابد. اگر این بار R_H اکسپت کرد یعنی شکل نقض پیدا شده در غیر این صورت می‌توان مطمئن بود که شکل نقض نداریم

$U_{TM} = \{ \langle M, q \rangle \mid q \text{ is useless in } M \}$

زبان E_{TM} را کاهش می‌دهیم به U_{TM}. اگر فرض کنیم که زبان U_{TM} decidable است (فرض خلف)

حال فرض می‌کنیم که R این زبان را decide می‌کند. حال decider زبان E_{TM} را به این صورت می‌سازیم.

می‌دانیم اگر q یک state از نوع accepting باشد این وضعیت بی فایده خواهد بود و فقط اگر زبان ماشین تورینگ M تهی باشد، حال اگر برای E_{TM}،



(E_{TM} مشخص می‌کند M زبانی تهی است یا خیر)

در S ما q_{acc} را از M در کنار M قرار داده

و به ماشین R می‌دهیم. اگر R $\langle M, q_{acc} \rangle$ را بپذیرد یعنی E_{TM} دل را نپذیرفته و اگر R نپذیرد یعنی E_{TM} آن را نپذیرفته. با توضیحات فوق توانستیم E_{TM} را decide کنیم که این تناقض است! از این تناقض نتیجه می‌گیریم فرض خلف نادرست بوده پس U_{TM} یک زبان تصمیم ناپذیر است ✓

۸) اگر BB قابل محاسبه باشد یعنی یک TM برای محاسبه کردن آن وجود دارد که آن را M می‌نامیم .

M روی ورودی 1^n (n۱) ، متوقف می‌شود ! ($1^{BB(n)}$ روی نوار) از برهان خلف استفاده می‌کنیم .

حال اگر بخواهیم با روش دوم ماشینی در نظر بگیریم این ماشین باید در نظر گرفتن M و یافتن و حرکت شروع از نوارهای M ـ یک state ، از نوع accept یا reject برسد ! اکنون این ماشین را M' در نظر بگیریم

M' ابتدا n تا ۱ را روی نوارهای M می‌نویسد . و سپس تعداد ۱ ها را به 2n تا می‌رساند و سپس M را روی این 2n تا ۱ اجرا می‌کند یعنی M با 1^{2n} مواجه است ! سپس M' با همان $BB(2n) \geq 1$ که دارد وقتی با نوارهای شروع کند حتما متوقف می‌شود . در M' برای ساخت n تا ۱ ـ n state و برای M نیز m استیت نیاز است !

سپس بیشترین تعداد ۱هایی که یک ماشین تورینگ با این تعداد state می‌سازد و در آن halt می‌کند همان $BB(n+m)$ است . همچنین برای تمامی n ها $BB(2n) \leq BB(n+m)$ است

اما از طرفی BB یک تابع صعودی است و در واقع می‌توان n هایی داشت که از m بزرگتر باشند که نتیجه می‌شود $BB(n+m) < BB(2n)$! که این دو حالت با هم در تناقض هستند .

پس $BB(x)$ قابل محاسبه است بنابراین فرض خلف اشتباه است و نقض می‌شود !

یعنی $BB(x)$ غیر قابل محاسبه است ! (undecidable)

در واقع تابع BB غیر قابل محاسبه است چرا که با سرعت بیشتری از سرعت ماشین‌ها رشد می‌کند .